

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :	Yuh-Cherng Wu	Art Unit :	2174
Serial No. :	10/829,145	Examiner :	Boris M. Pesin
Filed :	April 21, 2004	Conf. No. :	5096
Title :	SOFTWARE CONFIGURATION PROGRAM FOR SOFTWARE APPLICATIONS		

Mail Stop Appeal Brief - Patents

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

SAP Aktiengesellschaft, the assignee of this application, is the real party in interest.

(2) Related Appeals and Interferences

The following cases have at least one common inventor and may be considered related appeals:

(1) App. No. 10/320,005 (Attorney Docket No. 13906-0083001):

An appeal brief was filed on November 13, 2006. Prosecution was reopened by the Examiner in a Feb. 21, 2007 non-final office action. A notice of abandonment was mailed June 2, 2008.

(2) Appeal 2009-010593, App. No. 10/918,219 (Attorney Docket No. 13906-0202001):

A decision on appeal dated Dec. 31, 2009 affirmed the Examiner's rejections of the claims. A notice of abandonment was mailed March 15, 2010.

(3) Appeal 2009-8464001, App. No. 10/365,671 (Attorney Docket No. 13906-0085001):

An Examiner's answer to the appeal brief was mailed Oct. 1, 2008. A docketing notice was received on April 23, 2009. The appeal is presently pending before the Board.

(3) Status of Claims

Claims 1-15 are pending. In the non-final office action dated January 22, 2010, claims 1-3, 5-8, 10-13, and 15 were rejected and claims 4, 9, and 14 were objected to as being dependent

upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claims 1, 6, and 11 are independent. A notice of appeal is filed herewith.

Applicant appeals the rejections of rejected claims 1-3, 5-8, 10-13, and 15.

(4) Status of Amendments

All amendments have been entered. (No amendments were made after the January 22, 2010, non-final office action or the previous June 4, 2009, final office action.) A listing of the current claims is provided in the Appendix provided with this Appeal Brief.

(5) Summary of Claimed Subject Matter

Independent claim 1 is directed to a method to be executed as part of a process for creating an executable configuration program. For example, the disclosure relates to “creating and executing a software configuration program for configuring software applications.” (Spec. 1:5-7.) The software configuration can include “multiple steps that are successively executed.” As an illustration, configuration of a complex software application may include the execution of a program (e.g., a “wizard” or “assistant”) that includes multiple, sequential steps for properly configuring the application. (Spec. 2:13-16; 13:18 to 14:8; 14:13-15.) Each of the steps can be associated with user-selectable options. For example, in a first step of the configuration program a user may select from five options, and in a second step of the configuration program a user may select from three options. (Spec. 13:18 to 14:8.)

The method recites steps of “generating a user interface,” “creating and storing in a repository [a] rule,” and “binding the rule in the repository to [a] user-selectable option.” Regarding the “generating a user interface step,” claim 1 states that the generated user interface includes a “logic flow area for a user to define a command structure for the configuration program including at least one step.” As an illustration, FIG. 4 of the specification shows a diagram of an example user interface for designing a configuration software application. (Spec. 5:21-22.) The user interface includes a logic flow area 84 that allows a user to arrange an ordering of steps to be presented to a user in the software configuration application. (Spec. 19:9-19; FIG. 4.) Claim 1 states that the user interface includes “a refinement area for the user to

specify a configuration detail regarding a step arranged in the logic flow area.” As an illustration, the graphical user interface shown in FIG. 4 includes a refinement area that can be activated for a particular step when a step is dragged from the flow chart palette 82 to the logic flow area 84. Once activated, the refinement area can allow a user to specify configuration details for a step, for example, explanatory text 92 that is displayed to a user of the executable configuration program. (Spec. 19:20 to 20:6; FIG. 4.) Claim 1 states that the user interface also includes “a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields.” For example, the graphical user interface shown in FIG. 4 includes a rule palette 86 that allows a user to selected conditional operators (e.g., “If,” “And,” “Else,” “Or,” etc.) and entry fields (e.g., input areas where user selectable options can be specified) that can be arranged together to express rules that are executed by the software program during runtime. (Spec. 9:9-22; FIG. 4.)

Regarding the “creating and storing in a repository [a] rule step,” claim 1 states recites a rule “that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.” For example, a rule can determine, for user-selected options in a first graphical user interface of a configuration program, the user-selected options that will be displayed in a next graphical user interface of the configuration program. (Spec. 11:1-11.) The rules can be stored in a data repository 24. (Spec. FIG. 1; 16:22 to 17:6.) Claim 1 states that “the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area.” As shown in example graphical user interface 24, steps can be arranged in the logic flow area and an ordering between them defined by the connectors 95. (Spec. 19:9-19; FIG. 4.) The rule can be defined in the refinement area. (Spec. 19:20 to 20:6; FIG. 4.)

Regarding the “binding the rule in the repository to [a] user-selectable option” step, claim 1 states that the rule is bound to the repository “so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.” For example, when a user makes a selection in a first graphical user interface of the configuration program, a rule defined in the refinement area 88 in the user interface for creating the configuration program may be executed. (Spec. 16:7-10.)

Independent claim 6 is directed to a system. The system includes a “computer network,” and a “service delivery device coupled to the network.” Claim 6 states that “the service delivery device include[es] a processor and memory storing instructions that, in response to receiving a first type of request for access to a service, cause the processor to” perform certain operations. For example, a system 10 includes a processor 12 and random access memory 13, and is connected to a network 19 (*See* FIG. 1.) The operations will “create an executable program,” “generated a user interface,” “create and store in a repository a rule,” and “bind the rule in the repository.” Regarding the “create an executable configuration program” step, claim 6 states that a configuration program may be created “program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options.” As an illustration, configuration of a complex software application may include the execution of a program (e.g., a “wizard” or an “assistant”) that includes multiple, sequential steps for properly configuring the application. (Spec. 2:13-16; 13:18 to 14:8; 14:13-15.) Each of the steps can be associated with user-selectable options. For example, in a first step of the configuration program a user may select from five options, and in a second step of the configuration program a user may select from three options. (Spec. 13:18 to 14:8.)

Regarding the “generate a user interface step,” claim 6 states that the generated user interface includes a “logic flow area for a user to define a command structure for the configuration program including at least one step.” As an illustration, FIG. 4 of the specification shows a diagram of an example user interface for designing a configuration software application. (Spec. 5:21-22.) The user interface includes a logic flow area 84 that allows a user to arrange an ordering of steps to be presented to a user in the software configuration application. (Spec. 19:9-19; FIG. 4.) Claim 6 states that the user interface includes “a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area.” As an illustration, the graphical user interface shown in FIG. 4 includes a refinement area that can be activated for a particular step when a step is dragged from the flow chart palette 82 to the logic flow area 84. Once activated, the refinement area can allow a user to specify configuration details for a step, for example, explanatory text 92 that is displayed to a user of the executable configuration program. (Spec. 19:20 to 20:6; FIG. 4.) Claim 6 states that the user interface also includes “a rule palette for the user to create a rule, wherein the rule palette provides multiple

conditional operators and entry fields.” For example, the graphical user interface shown in FIG. 4 includes a rule palette 86 that allows a user to selected conditional operators (e.g., “If,” “And,” “Else,” “Or,” etc.) and entry fields (e.g., input areas where user selectable options can be specified) that can be arranged together to express rules that are executed by the software program during runtime. (Spec. 9:9-22; FIG. 4.)

Regarding the “create and store in a repository [a] rule step,” claim 6 states recites a rule “that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.” For example, a rule can determine, for user-selected options in a first graphical user interface of a configuration program, the user-selected options that will be displayed in a next graphical user interface of the configuration program. (Spec. 11:1-11.) The rules can be stored in a data repository 24. (Spec. FIG. 1; 16:22 to 17:6.) Claim 6 states that “the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area.” As shown in example graphical user interface 24, steps can be arranged in the logic flow area and an ordering between them defined by the connectors 95. (Spec. 19:9-19; FIG. 4.) The rule can be defined in the refinement area. (Spec. 19:20 to 20:6; FIG. 4.)

Regarding the “bind the rule in the repository to [a] user-selectable option” step, claim 6 states that the rule is bound to the repository “so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.” For example, when a user makes a selection in a first graphical user interface of the configuration program, a rule defined in the refinement area 88 in the user interface for creating the configuration program may be executed. (Spec. 16:7-10.)

Independent claim 11 is directed to a computer-readable storage device that can include instructions executed by a processor. For example, the system 10 can include non-volatile memory 14 that includes a design-time software program 16 used to generate a configuration software program and a run-time software program 18, as well as a processor 12. (See Spec. 7:11-17; FIG. 1.) Claim 11 states that when the instructions are executed by the processor, the processor will “create an executable configuration program,” “generate a user interface,” “create and store in a repository [a] rule,” and “bind the rule in the repository to the specified user-selectable option.”

Regarding the “create an executable configuration program” step, claim 11 states that a configuration program may be created “program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options.” As an illustration, configuration of a complex software application may include the execution of a program (e.g., a “wizard” or an “assistant”) that includes multiple, sequential steps for properly configuring the application. (Spec. 2:13-16; 13:18 to 14:8; 14:13-15.) Each of the steps can be associated with user-selectable options. For example, in a first step of the configuration program a user may select from five options, and in a second step of the configuration program a user may select from three options. (Spec. 13:18 to 14:8.)

Regarding the “generate a user interface step,” claim 11 states that the generated user interface includes a “logic flow area for a user to define a command structure for the configuration program including at least one step.” As an illustration, FIG. 4 of the specification shows a diagram of an example user interface for designing a configuration software application. (Spec. 5:21-22.) The user interface includes a logic flow area 84 that allows a user to arrange an ordering of steps to be presented to a user in the software configuration application. (Spec. 19:9-19; FIG. 4.) Claim 11 states that the user interface includes “a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area.” As an illustration, the graphical user interface shown in FIG. 4 includes a refinement area that can be activated for a particular step when a step is dragged from the flow chart palette 82 to the logic flow area 84. Once activated, the refinement area can allow a user to specify configuration details for a step, for example, explanatory text 92 that is displayed to a user of the executable configuration program. (Spec. 19:20 to 20:6; FIG. 4.) Claim 11 states that the user interface also includes “a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields.” For example, the graphical user interface shown in FIG. 4 includes a rule palette 86 that allows a user to selected conditional operators (e.g., “If,” “And,” “Else,” “Or,” etc.) and entry fields (e.g., input areas where user selectable options can be specified) that can be arranged together to express rules that are executed by the software program during runtime. (Spec. 9:9-22; FIG. 4.)

Regarding the “create and store in a repository [a] rule step,” claim 11 states recites a rule “that during execution of the configuration program determines which of the user-selectable

options for the step are displayed when a specified user-selectable option of a previous step is selected.” For example, a rule can determine, for user-selected options in a first graphical user interface of a configuration program, the user-selected options that will be displayed in a next graphical user interface of the configuration program. (Spec. 11:1-11.) The rules can be stored in a data repository 24. (Spec. FIG. 1; 16:22 to 17:6.) Claim 11 states that “the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area.” As shown in example graphical user interface 24, steps can be arranged in the logic flow area and an ordering between them defined by the connectors 95. (Spec. 19:9-19; FIG. 4.) The rule can be defined in the refinement area. (Spec. 19:20 to 20:6; FIG. 4.)

Regarding the “bind the rule in the repository to [a] user-selectable option” step, claim 11 states that the rule is bound to the repository “so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.” For example, when a user makes a selection in a first graphical user interface of the configuration program, a rule defined in the refinement area 88 in the user interface for creating the configuration program may be executed. (Spec. 16:7-10.)

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1-3, 5-8, 10-13, and 15 are pending with claims 1, 6, and 11 being independent. The Office Action (at page 3) rejected claims 1-3, 6-8, and 10-13 under 35 U.S.C. § 102(e) as being anticipated by Bennett et al. (U.S. Pat. No. 6,782,528)¹. The Office Action (at page 5) rejected dependent claims 5, 10, and 15 under 35 U.S.C. § 103(a) as being unpatentable over Bennett in view of SAP Wizard Builder. The Office Action (at page 6) objected to dependent claims 4, 9, and 14 to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form. Applicant is appealing the rejection of pending claims 1-3, 5-8, 10-13, and 15.

I. Bennett does not disclose “creating and storing a rule that during execution of the configuration program determines which of the user-selectable options for the step are displayed

¹ Applicant notes that the Office Action appears to have erroneously cited to U.S. Pat. No. 6,782,518 (listing ‘Decker’ as inventor). Indeed, U.S. Pat. No. 6,782,528 is listed as invented by Bennett et al. and includes the figures and elements that were cited in the Office Action.

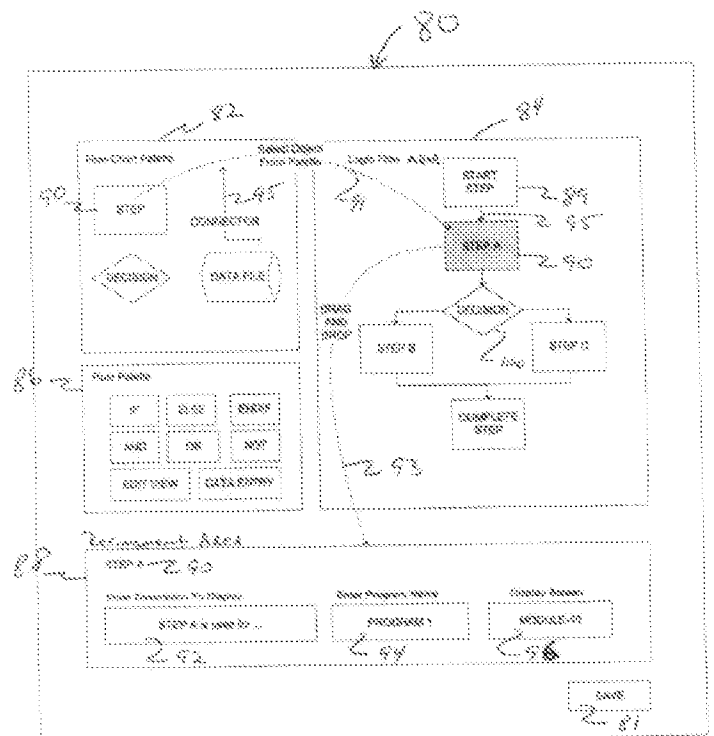
II. Bennett does not disclose “a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields” (independent claims 1, 6, and 11).

(7) Argument

For the following reasons, Applicant respectfully asserts that the present claims are patentable over the references of record, and request that the above rejections be reversed.

I. Rejection of claims 1-3, 5-8, 10-13, and 15 — Bennett does not disclose “creating and storing a rule that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.”

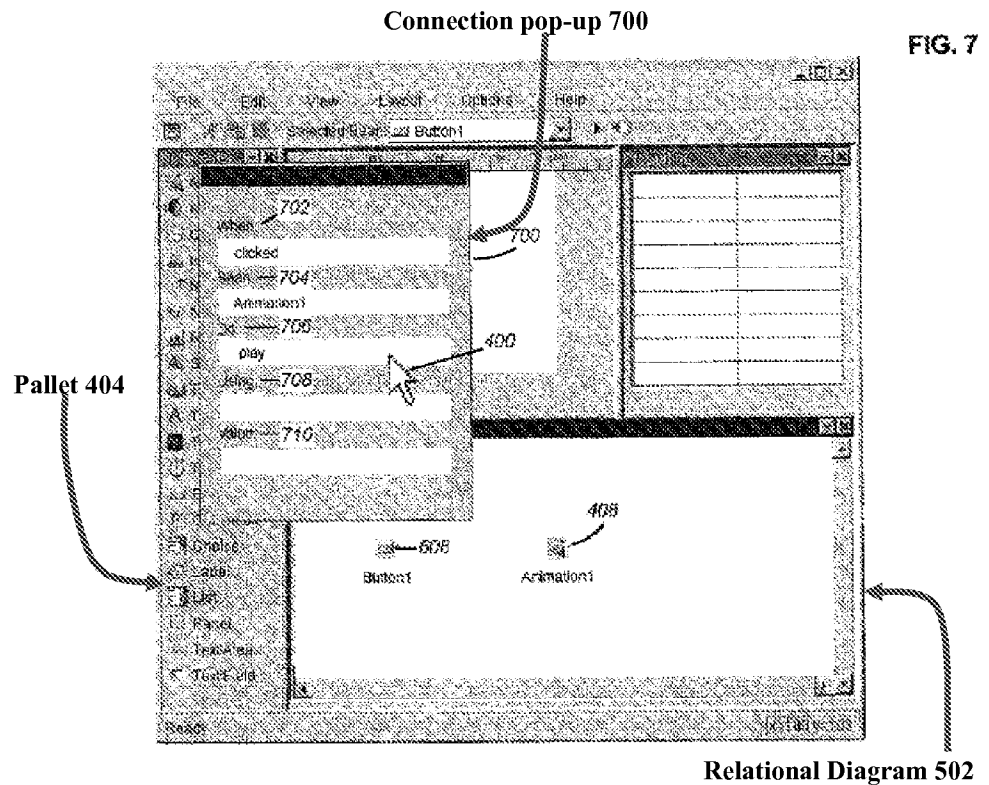
By way of introduction, the claimed subject matter can involve a method for creating an executable configuration program (e.g., a “wizard” for configuring programs). (Spec. 1:10-21.) The method can include providing a graphical interface that allows a user to create the software configuration program. (Spec. 1:10-21.) After the software configuration program has been created, execution of the configuration program can present to a user a series of screens. Each of the screens may present several options to a user for selection. (Spec. 2:13-16; 13:18 to 14:8; 14:13-15.) Selection of an option in a screen can impact user-options that are presented in a subsequently displayed screen. (Spec. 16:7-10.)



554

The graphical interface described in claims 1, 6 and 11 may be used to create the described “software configuration program.” The graphical interface includes “a logic flow area,” a “refinement area,” and a “rule palette.” (*See, e.g.*, Spec. FIG. 4, reproduced herein.) A user of the graphical interface can arrange “steps” in the logic flow area to define an flow of screens that are presented to a user when the configuration program is created. (Spec. 19:9-19; FIG. 4.) Upon user-selection of a step that is displayed in the logic flow area, a “refinement area” can be activated for the step. (Spec. FIG. 4; 19:20 to 20:6.) Once activated, the refinement area can allow a user to specify configuration details for a step, for example, explanatory text 92 that is displayed to a user of the executable configuration program. (*Id.*) The graphical user interface can also include a rule palette 86 that allows a user to selected conditional operators (e.g., “If,” “And,” “Else,” “Or,” etc.) and entry fields (e.g., input areas where user selectable options can be specified) that can be arranged together in the refinement area to express rules that are executed by the software program during runtime and in connection with a specific step. (Spec. 9:9-22; 21:13 to 22:12; FIG. 4.) After a user has arranged the “steps” and created the “rules” with the graphical user interface (e.g., the interface displayed in FIG. 4), a software configuration program can be created that successively executes the steps and displays user-selectable options in each step based on the rules that were created with the graphical interface.

Bennett teaches a user interface that allows an individual to “create applets” by defining “relationships between the beans contained within the applet.” (Bennett, at 2:66 to 3:5.) A user can drag a “bean icon” from the “pallet 404” (see FIG. 4) and drop the bean icon on the “relational diagram” 502 (see FIG. 5). (Bennett, at 3:42-56.)



A second bean may be dragged from the pallet to the relational diagram, and a “Connection pop-up 700” appears that enables a user to define a relationship between the two beans. (Bennett, at 3:63 to 4:14.) For example, the “When 702” field displays the “default event of the bean dropped in the left column.” The “Bean 704” field displays the “name of the bean in the center column.” The “Do 706” field displays the “default action of the bean in the center column.” Thus, in the illustration in FIG. 7, the “Connection pop-up 700” shows that the “Animation 1” bean will “play” when “Button1” bean is clicked. In other words, Bennett teaches a graphical interface for visually defining the relationships between various “beans” that are executed by an “applet.”

The Office Action (at pages 3-4) states:

Applicant agrees that “line 800”

displayed when a specified user-selectable option of a previous step is selected.” Even if it is assumed, for the sake of argument, that execution of the “applet” that includes the “Button1” and “Animation1” beans taught the recited “execution of the configuration program,” Bennett fails to teach a rule that determines – during execution of the applet – which user-selectable options are displayed for “Animation1” when a user-selectable option for the “Button1” step is selected. Indeed, when the “Animation1” bean is executed as part of the applet, no user selectable-options appear to be displayed. In other words, Bennett is silent on user-selectable options being displayed during execution of the purported configuration program (i.e., the “applet”).

For at least these reasons, Bennett fails to teach a rule that determines which user-selectable options are displayed when a specified user-selectable option of a previous step is selected. Applicant respectfully submits that claim 1 is patentable over Bennett and all other references in the record. Independent claims 6 and 11 contain similar language and are patentable for at least the same reasons. Dependent claims 2-3, 5, 7-8, 10, and 12-13, and 15 are patentable for at least the same reasons, and for the independently patentable features recited therein.

II. Rejection of claims 1-3, 5-8, 10-13, and 15 — Bennett does not disclose “a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields.”

Regarding independent claim 1, Bennett fails to teach a *“a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields.”* The Office Action (at page 3) contends that the pallet 404 (*see* FIG. 7, above) that includes a listing of “beans” that may be dragged onto the “relational diagram 502” teaches the recited rule pallet. In particular, the Office Action states:

(iii) a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields (See Figure 4, Element 404);

Applicant respectfully disagrees. The described palette “allows a user to create applets by first selecting beans from a palette” and dragging them over the “relational diagram 502.” (Bennett, 2:66 to 3:3 and 3:19-3:45.) In other words, the palette 404 that is relied on by the Examiner includes beans (i.e., the purported “steps”) but does not include “conditional operators and entry fields” for creating a rule “that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.” Indeed, the relationship between “Button1” and “Animation1” (the purported “rule”) is defined by the fields in the “Connection pop-up 700” (*see* FIG. 7 above). The relationship between “Button1” and “Animation1” is not defined using the palette 404.

For at least this reason, Bennett fails to teach a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields. Applicant respectfully submits that claim 1 is patentable over Bennett and all other references in the record. Independent claims 6 and 11 contain similar language and are patentable for at least the same reasons. Dependent claims 2-3, 5, 7-8, 10, and 12-13, and 15 are patentable for at least the same reasons, and for the independently patentable features recited therein.

Conclusion

Accordingly, the Bennett fails to teach the subject matter described in independent claims 1, 6, and 11. Applicant respectfully submits that independent claims claim 1, 6, and 11 are patentable over Simonsen and all other references in the record. Dependent claims 2-3, 5, 7-8, 10, and 12-13, and 15 are patentable for at least the same reasons as claim 4 and for the additional inventive combinations described therein. Allowance of claims 1-3, 5-8, and 10-13, and 15 is respectfully requested.

No fees are believed to be due as the fee for the notice of appeal and the brief on appeal filed on October 19, 2009 is being applied to this new appeal (*see* January 22, 2010 Office Action, at page 2). Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: April 8, 2010

/Andrew Dommer, reg. no. 65,758/
Andrew Dommer
Reg. No. 65,758

Fish & Richardson P.C.
3200 RBC Plaza
60 South Sixth Street
Minneapolis, Minnesota 55402
Telephone: (612) 335-5070
Facsimile: (877) 769-7945

Appendix of Claims

1. A method to be executed as part of a process for creating an executable configuration program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options, the method comprising:

generating a user interface including at least (i) a logic flow area for a user to define a command structure for the configuration program including at least one step, (ii) a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area, and (iii) a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields;

creating and storing in a repository the rule that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected, wherein the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area; and

binding the rule in the repository to the specified user-selectable option so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.

2. The method of claim 1 wherein the binding of the rule to the specified user-selectable option is performed by virtue of a designer selecting a user-selectable option for which to create the rule.

3. The method of claim 1 further comprising:

creating and storing in the repository a textual explanation of the step that describes available user-selectable options for the step; and

binding the textual explanation in the repository to the step so that during execution of the configuration program the textual explanation of the step is displayed.

4. The method of claim 3 wherein creating the textual explanation comprises translating the textual explanation into at least one different language.

5. The method of claim 1 further comprising evaluating the stability of a configured software application by executing the software application using a simulated user-selectable option.

6. A system comprising:

a computer network;

a service delivery device coupled to the network, the service delivery device including a processor and memory storing instructions that, in response to receiving a first type of request for access to a service, cause the processor to:

create an executable configuration program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options;

generate a user interface including at least (i) a logic flow area for a user to define a command structure for the configuration program including at least one step, (ii) a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area,

and (iii) a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields;

create and store in a repository the rule that during execution of the configuration program determines which of the user-selectable options for a step are displayed when a specified user-selectable option of a previous step is selected, wherein the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area; and

bind the rule in the repository to the specified user-selectable option so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.

7. The system of claim 6 wherein the memory stores instructions that, in response to receiving the first type of request, cause the processor to bind the rule to the specified user-selectable option by providing an ability to select a user-selectable option for which to create the rule.

8. The system of claim 6 wherein the memory stores instructions that, in response to receiving the first type of request, cause the processor to:

create and store in the repository a textual explanation of the step that describes available user-selectable options for the step; and

bind the textual explanation in the repository to the step so that during execution of the configuration program the textual explanation of the step is displayed.

9. The system of claim 8 wherein the memory stores instructions that, in response to receiving the first type of request, cause the processor to translate the textual explanation into at least one different language.

10. The system of claim 6 wherein the memory stores instructions that, in response to receiving the first type of request, cause the processor to evaluate the stability of a configured software application by executing the software application using a simulated user-selectable option.

11. A computer-readable storage device comprising instructions that, when executed by a processor, cause the processor to:

create an executable configuration program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options;

generate a user interface including at least (i) a logic flow area for a user to define a command structure for the configuration program including at least one step, (ii) a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area, and (iii) a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields;

create and store in a repository the rule that during execution of the configuration program determines which of the user-selectable options for a step are displayed when a specified user-selectable option of a previous step is selected, wherein the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area; and

bind the rule in the repository to the specified user-selectable option so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected.

12. The storage device of claim 11 including instructions that, when executed by the processor, cause the processor to bind the rule to the specified user-selectable option by providing an ability to select a user-selectable option for which to create the rule.

13. The storage device of claim 11 including instructions that, when executed by the processor, cause the processor to:

create and store in the repository a textual explanation of the step that describes available user-selectable options for the step; and

bind the textual explanation in the repository to the step so that during execution of the configuration program the textual explanation of the step is displayed.

14. The storage device of claim 13 including instructions that, when executed by the processor, cause the processor to translate the textual explanation into at least one different language.

15. The storage device of claim 13 including instructions that, when executed by the processor, cause the processor to evaluate the stability of a configured software application by executing the software application using a simulated user-selectable option.

Applicant : Yuh-Cherng Wu
Serial No. : 10/829,145
Filed : April 21, 2004
Page : 19 of 20

Attorney's Docket No.: 13906-0122001 / 2003P00271 US

Evidence Appendix

None.

Applicant : Yuh-Cherng Wu
Serial No. : 10/829,145
Filed : April 21, 2004
Page : 20 of 20

Attorney's Docket No.: 13906-0122001 / 2003P00271 US

Related Proceedings Appendix

None.